AD-A058 452 CORNELL UNIV ITHACA N Y DEPT OF COMPUTER SCIENCE F/G 12/1
THE DIRECTED SUBGRAPH HOMEOMORPHISM PROBLEM.(U)
JUN 78 S FORTUNE, J HOPCROFT, J WYLLIE N00014-76-C-0018
UNCLASSIFIED CU-CSD-TR-78-342 NL

1 OF 1
AD
A058452

END
DATE
FILMED
11-78
DDC

LEVEL II

12

B.S.

ADA058452

AD No. ___
DDC FILE COPY

6 THE DIRECTED SUBGRAPH HOMEOMORPHISM*
PROBLEM.

by

10 Steven Fortune,
John Hopcroft
James Wyllie

12 23p.

11 Jun 78

9 Technical rept.,

DDC
SEP 7 1978
E

14 TR-CSD-TR-78-342

Department of Computer Science
Cornell University
Ithaca, NY 14853

*This research was supported in part by the office of Naval Research under contract number N00014-76-C-0018.
15

78 08 29 031

407 072

THE DIRECTED HOMEOMORPHISM PROBLEM

By

Steven Fortune
John Hopcroft
James Wyllie

Department of Computer Science
Cornell University

Abstract

The set of pattern graphs for which the fixed directed subgraph homeomorphism problem is NP-complete is characterized. A polynomial time algorithm is given for the remaining cases. The restricted problem where the input graph is a directed acyclic graph is in polynomial time for all pattern graphs and an algorithm is given.

Introduction

The subgraph homeomorphism problem is to determine if a pattern graph P is homeomorphic to a subgraph of an input graph G. The homeomorphism maps nodes of P to nodes of G and arcs of P to simple paths in G. The graphs P and G are either both directed or both undirected. The paths in G corresponding to arcs in P must be pairwise node-disjoint. The mapping of nodes in P to nodes in G may be specified or left arbitrary.

This problem can be viewed as a generalized path-finding problem. For example, if the pattern graph consists of two disjoint arcs and the node mapping is given, then the problem is equivalent to finding a disjoint pair of paths between specified vertices in the input graph.

It is easy to see that the problem is NP-complete if it is posed as "Given a pair (P,G) as input, possibly with a node mapping specified, does G contain a subgraph homeomorphic to P?" This follows from the Hamilton circuit problem if the node mapping is unspecified and the results of Even, Itai and Shamir [2] on multi-commodity network flows if the node mapping is specified. LaPaugh and Rivest [4] discuss this in more detail.

We consider the question, for fixed pattern graph P, "Given as input a graph G with node-mapping specified, does G contain a subgraph homeomorphic to P?" We refer to this as the fixed subgraph homeomorphism problem. In this paper, under the assumption $P \neq NP$, we characterize the pattern graphs for which the fixed directed subgraph homeomorphism problem is NP-complete and for which pattern graphs it is polynomial time decidable. We also show that if the input graphs are

restricted to being directed and acyclic, then there is always a polynomial time algorithm. The general case of the undirected fixed subgraph homeomorphism problem remains open, although polynomial time algorithms are known for the pattern consisting of a cycle of length three [4] and the pattern of two disjoint edges [6].

Definitions

A directed graph G consists of a set N of nodes, a set A of arcs, and two functions head and tail mapping arcs to nodes. Given an arc a, we say that its head is the node head(a), or that a is incident to head(a). The tail of an arc and the expression "incident from" are defined analogously. We use this definition to allow graphs to have multiple parallel arcs as well as loops (a loop is an arc with identical head and tail). A path of length k from node x to node y is a sequence of arcs $(a_1, a_2, \ldots a_k)$ such that $x = \text{tail}(a_1)$, $y = \text{head}(a_k)$ and $\text{tail}(a_i) = \text{head}(a_{i-1})$ for $i = 2, \ldots k$. A path from x to y is simple if no node occuring as the head or tail of an arc is repeated, except that x may equal y. Two simple paths are node-disjoint if they have no nodes in common except that endpoints may be equal.

Given directed graphs P and G and a mapping m of the nodes of P into the nodes of G, we say P is homeomorphic to a subgraph of G if there exists a mapping from arcs of P to pairwise node-disjoint paths in G such that an arc with head h and tail t is mapped to a simple path from m(t) to m(h). The fixed subgraph homeomorphism problem, for fixed pattern graph P, is the problem

of determining on an input graph G and a node mapping m whether P is homeomorphic to a subgraph of G. We assume without loss of generality that every node in P has at least one incident arc.

We note that paths could be required to be pairwise arc-disjoint rather than node-disjoint. However, LaPaugh and Rivest [4] have shown that the two formulations are computationally equivalent for directed graphs.

The General Directed Case

Under the assumption that $P \neq NP$ we now characterize those directed pattern graphs for which the fixed subgraph homeomorphism problem is polynomial time decidable and those for which the problem is NP-complete. Let $\vec{C}$ be the collection of all directed graphs with a distinguished node called the root possessing the property that either the root is the head of every arc or the root is the tail of every arc. Note that the root may be both the head and tail of some arcs and thus loops at the root are allowed. Equivalently, a graph is in C if when all loops at the root are deleted and multiple arcs between pairs of nodes are merged into single arcs, the resulting graph is a tree of height at most one.

<u>Theorem 1</u>: For each P in C there is a polynomial time algorithm for the fixed subgraph homeomorphism problem with pattern P.

<u>Proof</u>: We will use the fact that finding maximum single-commodity flows in a directed network with node capacities is computable in polynomial time [1]. Suppose the pattern graph P is in C;

we will assume all arcs in P are directed away from the root. The case with the reverse direction is analogous. Also suppose we have an input graph G together with a mapping of the nodes of P to nodes of G.

We first note that if there are loops at the root of P, we can obtain an equivalent problem without loops as follows. We split the root of P into a new leaf and new root, with the loop arcs directed from the new root to the new leaf. All other edges incident from the old root are incident from the new root. In the input graph G we must now split the image of the old root into two nodes, one with all the incoming arcs and one with all the outgoing arcs. The new root in P is mapped to the node with outgoing arcs; the new leaf in P is mapped to the node with incoming arcs. Clearly, the original problem has a solution if and only if the new one does.

Now label the image of the root of P as a source with capacity equal to the outdegree of the root of P. Label the image of every other node in P as a sink with capacity equal to the indegree of the node in P. Give every unlabelled node in G capacity one, and every arc in G capacity one. Now decide if there is a flow in G equal to the capacity of the source. Clearly, since P is "tree-like", if P is homeomorphic to a subgraph of G, the flow exists. Conversely, if the flow exists then the condition that all non-source, non-sink nodes have capacity one guarantees that the arcs in P map to node-disjoint paths in G. □

Next we show that for each pattern P not in C the fixed subgraph homeomorphism problem with pattern P is NP-complete. We proceed with several lemmas.

Lemma 1 Suppose P is a subgraph of Q, and the subgraph homeomorphism problem is NP-hard with pattern P. Then it is NP-hard with pattern Q.

Proof Given a graph G together with a mapping g of nodes of P into nodes of G, we construct a graph H together with a mapping h of nodes of Q into nodes of H such that P is homeomorphic to a subgraph of G if and only if Q is homeomorphic to a subgraph of H.

Let Q-P be the graph consisting of arcs in Q not in P, together with incident nodes. Form H by adding to G a copy of Q-P, where a node n of Q-P also in the node set of P is identified with the node g(n) in G. Extend the mapping g to a mapping h from nodes of Q to nodes of H in the obvious way. If a is an arc in Q-P, then we denote by a' the corresponding arc in the copy of Q-P added to G.

Clearly, if P is homeomorphic to a subgraph of G then Q is homeomorphic to a subgraph of H. We show the converse by induction on the number of arcs in Q-P. This is vacuously true if Q-P is empty, so suppose Q-P is not empty and Q is homeomorphic to a subgraph of H. We first note that the image of any arc a in Q-P with at least one endpoint not in P can only have as its image the corresponding arc a' in the copy of Q-P. Hence no arc in P can have image containing an arc a' in the copy of Q-P, where arc a in Q-P has at least one endpoint not in P. Now if every arc a in P has image in G, then P is

homeomorphic to a subgraph of G. So suppose some arc p in P has image containing arc q' in the copy of Q-P. Arc q in Q-P must have both endpoints in P, hence q' must be the entire image of arc p, and arcs p and q are parallel in Q. Now if we change h so that the image of arc p is the image of arc q, and delete arc q from Q and q' from H, we have Q-{q} homeomorphic to a subgraph of H-{q'}. By the induction hypothesis, P is homeomorphic to a subgraph of G. □

Lemma 2: Consider the subgraph in Figure 1. Suppose there are two node-disjoint paths passing through the subgraph -- one leaving at node A the other entering at B. Then the path leaving at A must have entered at C and the path entering at B must leave at D. Further, there is exactly one additional path through the subgraph and it is either 8→9→10→4→11 or 8'→9'→10'→4'→11' depending on the actual routing of the path leaving at A.

Proof: Consider the path leaving at A, call it the "A-path". It must use either arc 1 or arc 1'. Since the subgraph is symmetric, assume it uses arc 1. Thus it must also use arc 2. The path entering at B, call it the "B-path" cannot use arc 6, hence it must use arc 6' and arc 2'. It cannot use arc 1', so it must use arc 7' and arc 9. The A-path cannot use arc 6, so it must use arcs 3 and 4. It cannot use arc 10, so it must use arc 5 and enter at C. The B-path cannot use arc 10 so it must use arc 12 and leave at D. The path 8→9→10→4→11 is now blocked and 8'→9'→10'→4'→11' is free. Notice that if a path enters at 8', it must leave at 11' as arcs 3' and 12' are blocked. Similarly, if a path leaves at 11' it must enter at 8'. □

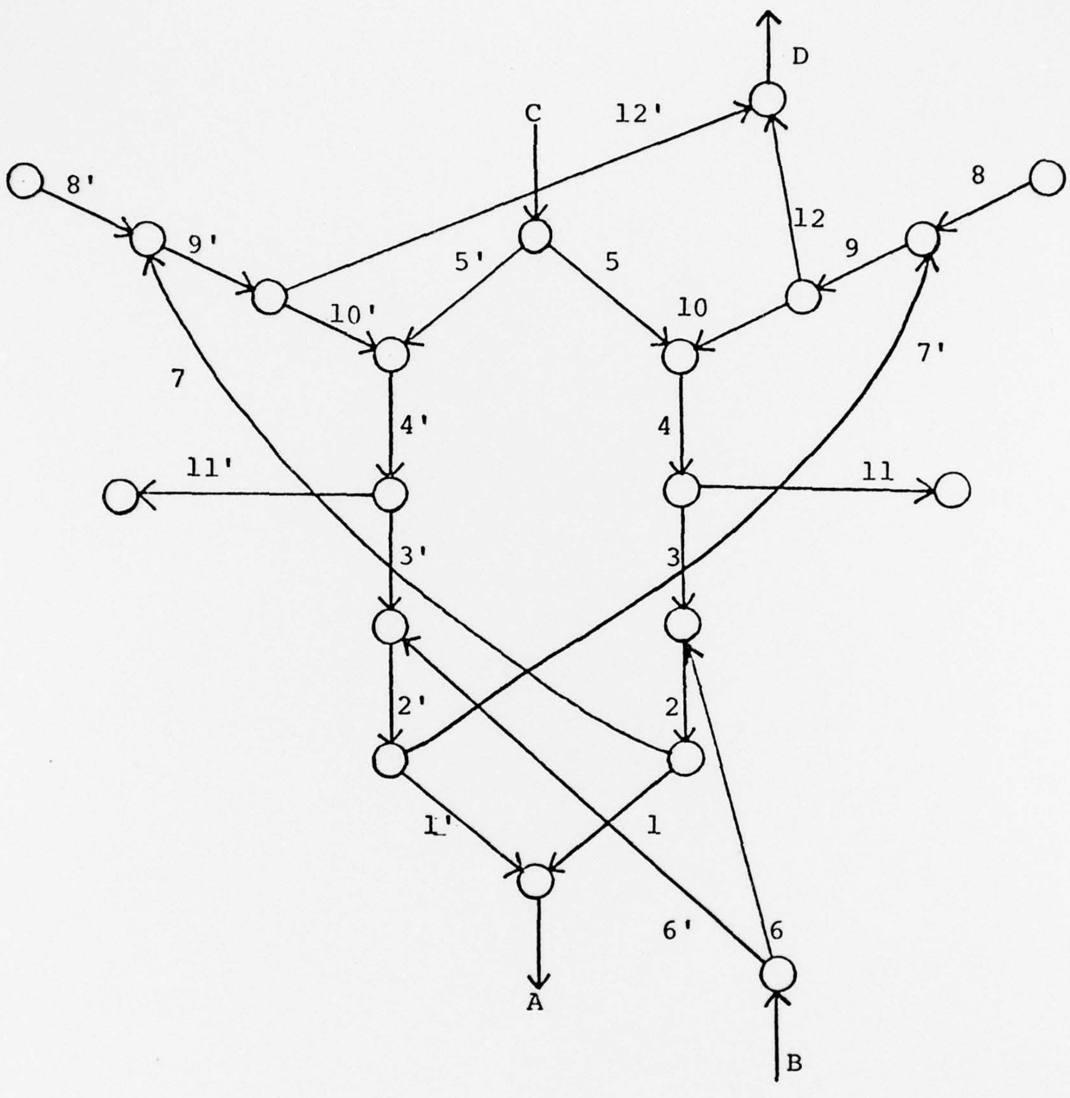

Figure 1. A Switch

We call the subgraph of Figure 1 a switch. We can stack arbitrarily many switches and still have the lemma apply by merging the C and D arcs of one switch with the A and B arcs of the next switch, respectively. A switch is represented schematically in Figure 2, where the vertical arcs represent the paths $8 \to 9 \to 10 \to 4 \to 11$ and $8' \to 9' \to 10' \to 4' \to 11'$ and the horizontal line, not an arc, indicates that at most one of the vertical arcs can be used. The A- and B-paths are implicit in Figure 2.

Lemma 3: Let P consist of two disjoint directed arcs and the four incident vertices. Then the fixed SHP with pattern P is NP-hard.

Proof: We will reduce the satisfiability problem for Boolean formulas in 3-CNF to the subgraph homeomorphism problem with pattern P. Fix a formula F with variables $x_1 \ldots x_k$ and clauses $t_1 \ldots t_\ell$. We construct a graph G_F as follows.

For each variable x_i make a copy of the subgraph appearing in Figure 3. We associate one column of vertical arcs with the literal x_i, the other with $\bar{x}_i$. The number of arcs in each column is the number of occurrences of its associated literal in F. The subgraphs are stacked by connecting the bottom node of the subgraph for x_i to the top node of the subgraph for x_{i+1} by an arc. There are also nodes $n_0 \ldots n_\ell$ corresponding to the clauses $t_1 \ldots t_\ell$ of F, with three arcs directed from n_i to n_{i+1} for each i. There is also an arc from the bottom of the subgraph of x_k to n_0.

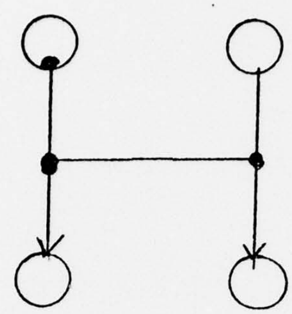

Figure 2. Schematic representation of a switch.

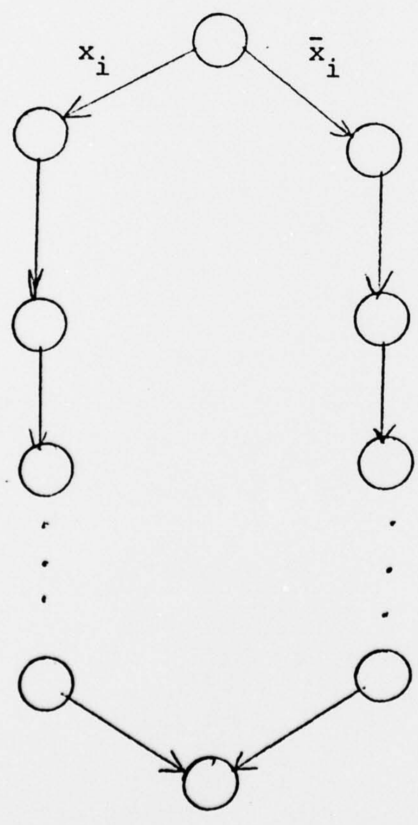

Figure 3

Now for each literal y appearing in each clause t_i we replace one of the arcs between n_{i-1} and n_i and one of the arcs in the column associated with y by a switch. The switches are linked together as described in the discussion after lemma 2. Finally we add nodes labelled W, X, Y and Z. The arc from Y is identified with the B input arc of the first switch, the arc from the D output of the last switch is connected to the top node of the subgraph for x_1, and there is an arc from n_ℓ to Z. The C input arc of the last switch is connected to W and the A output arc of the first switch is connected to X. An example of G_F is shown in Figure 4.

We claim there are node-disjoint paths from W to X and from Y to Z in G_F if and only if the formula F is satisfiable. Suppose F is satisfiable. Then the path from Y to Z can go through the column associated with $\bar{y}$ if y is true in the satisfying assignment. Then since at least one literal in each clause t_i is satisfied, there will always be at least one switch path usable from n_{i-1} to n_i. Conversely, if node-disjoint paths exist they must pass through the switches as described in Lemma 2. Hence the Y to Z path must proceed through the subgraphs for the x_i's and through nodes n_0 to n_ℓ. The assignment realized by setting literal y to be true if and only if the Y to Z path uses the column associated with $\bar{y}$ must satisfy F. This reduction from 3-CNF satisfiability to the fixed SHP is computable in polynomial time, hence the fixed SHP with pattern P is NP-hard. $\square$

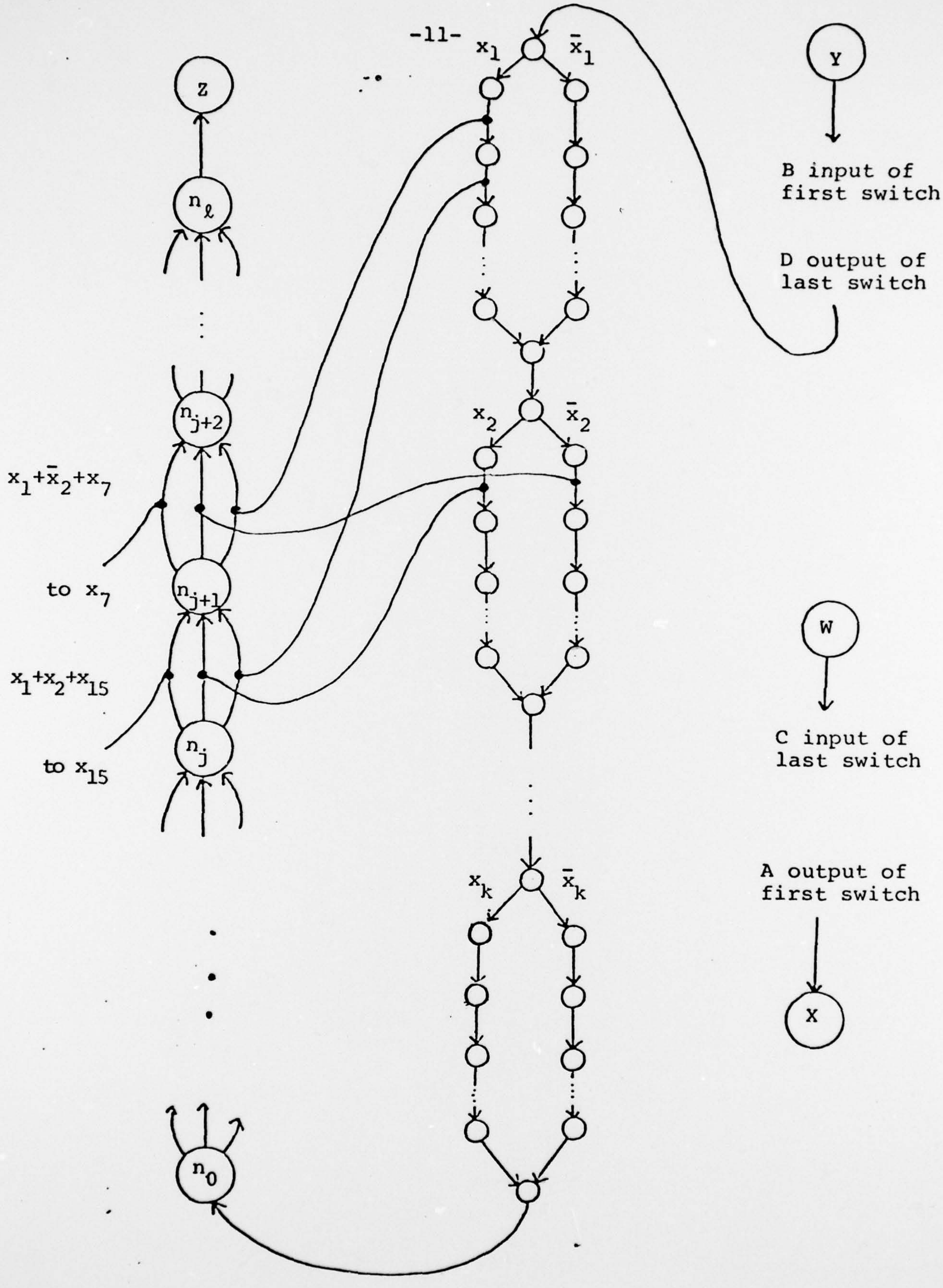
Z
n_ℓ
n_{j+2}
$x_1+\bar{x}_2+x_7$
to x_7
n_{j+1}
$x_1+x_2+x_{15}$
to x_{15}
n_j
n_0
x_1
$\bar{x}_1$
x_2
$\bar{x}_2$
x_k
$\bar{x}_k$
Y
B input of
first switch
D output of
last switch
W
C input of
last switch
A output of
first switch
X

Figure 4.

<u>Theorem 2</u>: For each P not in G the fixed subgraph homeomorphism problem with pattern P is NP-complete.

<u>Proof</u>: The fixed SHP for any pattern graph P is clearly in NP, so we need only show that for $P \notin C$, the problem is NP-hard. An alternative characterization of C is that a graph G is not in C if and only if G contains one of the following subgraphs:

i) two disjoint edges, one or both of which may be a loop,

ii) a path of two arcs visiting three distinct vertices, or

iii) a cycle of length two.

By showing that the fixed SHP for each of the above three subgraphs is NP-hard and then by applying Lemma 1, the theorem is established for all pattern graphs containing one of these graphs as a subgraph and hence for all graphs not in C. Lemma 3 establishes the NP-hardness of subgraph (i) in the case that there are no loops. If there are loops, identifying W with X and/or Y with Z allows the same construction to be used. For case (ii), identifying X and Y establishes the theorem, and finally in case (iii), identifying the pairs of vertices W, Z and X, Y allows the proof of Lemma 3 to carry over to this case. □

<u>Directed Acyclic Graphs</u>

In this section we show that for any fixed pattern graph the directed subgraph homeomorphism problem for acyclic input graphs has a polynomial time algorithm. The degree of the

polynomial depends on the particular pattern graph. The algorithm works whether or not the node mapping of pattern to input graph is specified. The result is a generalization of Perl and Shiloach's algorithm [5] for finding two node-disjoint paths in a directed acyclic graph.

Fix a pattern graph and assume for the moment that the mapping of the nodes of the pattern graph to nodes of the input graph is specified with the input graph. The algorithm is described in terms of a pebbling game played on the nodes of the input graph. Pebbles will correspond to the arcs of the pattern graph; the path traced by a pebble during the game will be the image of an arc in the pattern graph.

We define the <u>level</u> of a node in the input graph to be the length of a longest path in the graph from the node. Clearly, if there is a path from v to w, then the level of v is greater than the level of w.

The rules of the pebbling game are as follows:

(1) For each arc a_i in the pattern graph there is a pebble p_i. Initially, for each node s in the pattern graph, the pebbles corresponding to arcs leaving s are placed on the image of s in the input graph.

(2) At any step pebble p_i may be moved along a directed arc from n to m if
   (a) n has the highest level of any pebbled node. (If two pebbles are on nodes of equal, highest level either may be moved), and
   (b) either m has no pebble on it,

and (c) m is not the image of any node in the pattern graph, except possibly the head of a_i

(3) Pebble p_i may be removed from the graph if it is placed on the image of the head of a_i.

The game is won if all pebbles can be removed from the input graph.

Lemma 4: The pebbling game can be won if and only if the pattern graph is homeomorphic to a subgraph of the input graph.

Proof: First suppose there is a winning strategy. Clearly the sequence of arcs traversed by pebble p_i is a path from the image of the tail of a_i to the image of the head of a_i. We need to show that all the paths are node-disjoint, except of course for endpoints. Suppose the paths of pebbles p_i and p_j intersect at a node m which is not the endpoint of path i. Node m is not the image of a node in the pattern graph by condition (2c). Without loss of generality we can assume pebble p_j visits m first. By condition (2b), pebble p_j must leave m before pebble p_i arrives. But this contradicts condition (2a), as the level of the node on which p_i resides must be higher than the level of node m, on which p_j resides. Hence all paths are node-disjoint.

Conversely, assume that the pattern graph is homeomorphic to a subgraph H of the input graph. Number every arc in H by the level of its tail in the input graph. It is easy to see that repeatedly executing the following strategy wins the pebbling game. Choose a highest numbered arc a in H, move pebble p_i along a where p_i is chosen so that the image of arc a_i contains a, and delete a from H. If p_i is now placed on the image of the head of arc a_i, remove p_i from the graph.

<u>Theorem 3</u> For any fixed directed pattern graph P, there is a polynomial time algorithm to decide if a directed acyclic graph G contains a subgraph homeomorphic to P.

<u>Proof</u> We first assume the node mapping is specified. Suppose P has k arcs. For an input graph with n nodes, there are $(n+1)^k$ ways of putting up to k pebbles on the graph, hence at most as many configurations of the pebbling game. A polynomial time algorithm can construct a graph G' where nodes correspond to configurations and arcs to legal moves. A path finding algorithm can then decide if there is a path from the node corresponding to the starting configuration to the node of the winning configuration.

If node mappings are not given, the above algorithm can be run for all $\binom{n}{s}$ possible mappings where s is the number of nodes in the pattern graph. □

We note that the result of Even, Itai and Shamir [2] on multicommodity flows implies that the directed subgraph homeomorphism problem is NP-complete if both pattern and input graphs are given as input, even if the input graph is acyclic.

<u>Conclusions</u> We have characterized the complexity of the fixed directed subgraph homeomorphism problem for all pattern graphs. However, many questions remain open. One obvious one is the problem for undirected graphs. We do not know how to construct a "switch", as in Lemma 2, to prove the problem NP-complete. It is conceivable that there are polynomial time algorithms for all undirected pattern graphs, with the polynomial

depending on the pattern. LaPaugh and Rivest [4] have given a polynomial time algorithm for the pattern consisting of a cycle of length three; Shiloach [6] has given a polynomial time algorithm for the pattern of two disjoint edges. The problems for the corresponding directed patterns are NP-complete.

Another possible question is to study other restricted classes of input graphs. For example, the question of whether the fixed directed subgraph homeomorphism problem for planar graphs is NP-complete is open.

If we consider the subgraph homeomorphism problem when node mappings are not given, that is, when we are to find a homeomorphic image of the pattern graph anywhere in the input graph, the problem is still NP-complete. This follows since we can effectively label vertices in both input and pattern graphs by giving them high degree. In fact, it is amusing to note that testing for the presence of the subgraph in Figure 5a is NP-complete, while the subgraph of Figure 5b is absent if and only if the graph is reducible, a condition which can be tested efficiently [3]. An interesting question is whether the directed subgraph homeomorphism problem without node mappings is NP-complete when all nodes in both pattern and input graphs have either indegree 1 and outdegree 2 or indegree 2 and outdegree 1.

Alternatively one could study collections of patterns. Testing for the presence of the subgraph in Figure 6a or testing for the presence of the subgraph in Figure 6b are both NP-complete problems. Nevertheless if we don't care which subgraph

is present there is a polynomial time algorithm. Conceivably in the undirected, unlabelled case, determining if a specific Kuratowski subgraph is present is NP-complete even though there is a polynomial planarity testing algorithm.

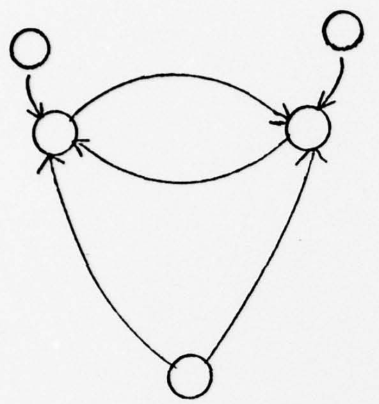

Figure 5a

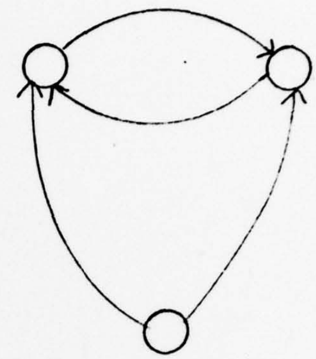

Figure 5b

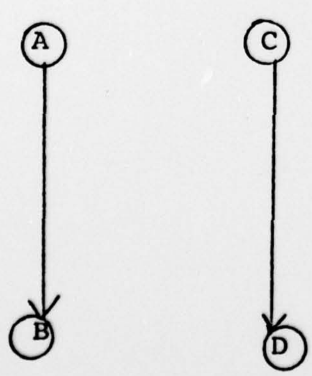

Figure 6A

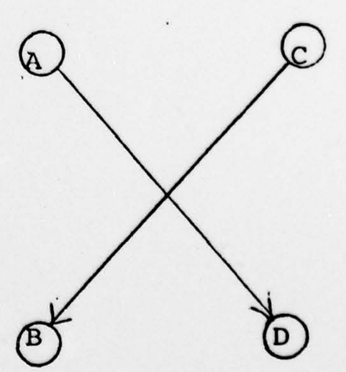

Figure 6B

References

1. Dinic, E.A. "Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation," Soviet Mathematics Doklady, 2:5 1970, pg. 1277-1280.

2. Even, S., Itai, A., Shamir, A. "On the Complexity of Timetable and Multi-commodity Flow Problems," SIAM J. on Computing 5:4, December 1976, pg. 691-703.

3. Hecht, Matthew, S., Ullman, Jeffrey D. "Flow Graph Reducibility," SIAM J. on Computing 1:2, June 1972, pg. 188-202.

4. LaPaugh, Andrea S., Rivest, Ronald L. "The Subgraph Homeomorphism Problem," Proc. Tenth Annual ACM Symposium on Theory of Computing, San Diego 1978, pg. 40-50.

5. Perl, Y., Shiloach, Y. "Finding Two Disjoint Paths Between Two Pairs of Vertices in a Graph," JACM 25:1, January 1978, pg. 1-9.

6. Shiloach, Y. Communication via reference [4].

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Computer Science Department Cornell University Ithaca, NY 14853	2a. REPORT SECURITY CLASSIFICATION 2b. GROUP

3. REPORT TITLE

The Directed Subgraph Homeomorphism Problem

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

technical

5. AUTHOR(S) (First name, middle initial, last name)

Steve Fortune, John Hopcroft, James Wyllie

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1978	21	6

8a. CONTRACT OR GRANT NO. N00014-76-C-0018	9a. ORIGINATOR'S REPORT NUMBER(S) TR78-342
b. PROJECT NO.	
c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) none

10. DISTRIBUTION STATEMENT

Distribution of manuscript is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

The set of pattern graphs for which the fixed directed subgraph homeomorphism problem is NP-complete is characterized. A polynomial time algorithm is given for the remaining cases. The restricted problem where the input graph is a directed acyclic graph is in polynomial time for all pattern graphs and algorithm is given.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
subgraph homeomorphism NP-complete directed acyclic graphs directed graphs						